Exhibit 20

Wayne Berry vs. Fleming Companies, Inc.

Expert Witness Report

Philip Johnson, Ph.D.

February 25, 2005

1. Statement of opinions to be expressed.

In the Spring of 2002, I participated as an expert witness in a trial concerning copyright infringement of software created by Wayne Berry. In December, 2004, I was again contacted by Timothy Hogan and Wayne Berry and asked to participate in a new case regarding this software.

In this report, I express the following opinion:

1. Fleming was using a derivation of the Berry database structure in their new version of the software.

2. Data considered in forming my opinion.

First, it is important to explain what it means to be a "derivation" in the case of database software. When defining a database, one typically creates a set of "tables". Each "row" in a table contains a set of data describing some aspect of the application. So, for example, one might have a "Customer" table in a database, where each row in the table provides data corresponding to one customer:

John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333,
Jane Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333
Kimo Smith, 23 Aloha Lane, Honolulu, HI 96822, 808 947-2222

So here we have a table, where each row contains data concerning one customer. However, the data is not organized very well: it would be nice to be able to extract the "Address" corresponding to a specific customer. So, a database table also contains columns, which allow you to characterize each of the kinds of data the comprising one row:

John Doe	46 Main St	Honolulu, HI 96822	808 956-3333
Jane Doe	46 Main St	Honolulu, HI 96822	808 956-3333
Kimo Smith	23 Aloha Lane	Honolulu, HI 96822	808 947-2222

A database typically contains many tables. Let's assume there is another table that stores "Orders" in terms of "Customers" and "Products":

EXHIBIT "E"

Exhibit 20

EXHIBIT Id
Johnson
55
5/19/05

John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333,	Samsung VCR Model 555 for $299.95 minus 15% employee discount
John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333,	Hitachi VCR Model 2221 for $668
Kimo Smith, 23 Aloha Lane, Honolulu, HI 96822, 808 947-2222	Sony VCR Model 321 for $999 on sale for $888.

There are several problems with this "Orders" table. First, there is redundancy: the information about John Doe is replicated. This can be a problem if, for example, John Doe changes his address. Second, just like with the first table, we would like to organize the data into more columns (so we could separate the List Price from Sale prices or other discounts, for example). Finally, you can see that the same information (i.e. about John Doe) appears in more than one table.

The task of the Database Designer is to look at all of the information required for storage, and to determine how to organize that data into tables such that (a) information is easy to change, (b) information is not duplicated as much as practical, (c) retrieval of information is fast, (d) information can be stored compactly. (Some of these goals are actually in conflict in practice (such as c and d), and so the Database Design must make trade-offs).

Going back to our example, one improvement would be to create a set of "ID" fields for the customer table, and then use those "IDs" instead of including the entire customer information in the Orders table:

Cust-001	John Doe	46 Main St	Honolulu, HI 96822	808 956-3333
Cust-002	Jane Doe	46 Main St	Honolulu, HI 96822	808 956-3333
Cust-003	Kimo Smith	23 Aloha Lane	Honolulu, HI 96825	808 947-2222

Cust-001	Samsung VCR Model 555 for $299.95 minus 15% employee discount
Cust-001	Hitachi VCR Model 2221 for $668
Cust-003	Sony VCR Model 321 for $999 on sale for $888.

These ID fields are called "pointers": they point to the information of interest. "Cust-001" is not the customer—it's a pointer to the customer John Doe. The first table contains a definition of the Customer pointer. The second table contains a reference to the Customer pointer.

At this point, you might thinking, "Hey, database design is easy, just look for duplicate information and make a pointer for it." Unfortunately, it's not that simple. For example, should we create a table called "City" so that we can get rid of the duplicate text in the City field?

City-001	Honolulu, HI 96822
City-002	Honolulu, HI 96825

Cust-001	John Doe	46 Main St	City-001	808 956-3333
Cust-002	Jane Doe	46 Main St	City-001	808 956-3333
Cust-003	Kimo Smith	23 Aloha Lane	City-002	808 947-2222

What about the duplicate phone number? What about the duplicate street address?

The crucial point is that when designing a database, deciding what tables to create, deciding how to organize the data into fields, and most importantly, deciding what data should be provided with a pointer and referred to elsewhere (as opposed to simply copying the data) requires skill and expertise. In the above example, we would almost certainly make a pointer for Customer, and would almost certainly not make a pointer for Telephone Number, even though both of those data values are duplicated.

In the Posner case, a distinction is made between "data" and "structure" with respect to copyright. The "data" cannot be copyrighted, only the structure can be copyrighted. In our example, the data (things like "John Doe") is not subject to copyright, while the structure (i.e. the decision to create a Customer table and an Orders table but not a City table or a Zip Code table) is copyrightable.

So, let's say that you have a database and you want to get the data out of it (which is not copyrightable) without taking any of the structure (which is copyrightable). That is very straightforward. What you do is create a single file that contains (a) the contents of each row of each table, where all of the fields have been concatenated together, and (b) where all of the pointers have been replaced by their corresponding data (after concatenation). That process removes the table structure, the field structure, and the pointer structure while preserving all of the data in the database.

Here's how it would work in our example. Take the following two tables which combine copyrightable structure with noncopywritable data:

Cust-001	John Doe	46 Main St	Honolulu, HI 96822	808 956-3333
Cust-002	Jane Doe	46 Main St	Honolulu, HI 96822	808 956-3333
Cust-003	Kimo Smith	23 Aloha Lane	Honolulu, HI 96825	808 947-2222

Cust-001	Samsung VCR Model 555 for $299.95 minus 15% employee discount
Cust-001	Hitachi VCR Model 2221 for $668
Cust-003	Sony VCR Model 321 for $999 on sale for $888.

Remove the internal field structure:

John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333,
Jane Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333
Kimo Smith, 23 Aloha Lane, Honolulu, HI 96822, 808 947-2222

Cust-001, Samsung VCR Model 555 for $299.95 minus 15% employee discount
Cust-001, Hitachi VCR Model 2221 for $668
Cust-003, Sony VCR Model 321 for $999 on sale for $888.

Now replace pointers by their associated data:

John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333,
Jane Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333
Kimo Smith, 23 Aloha Lane, Honolulu, HI 96822, 808 947-2222

John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333, Samsung VCR Model 555 for $299.95 minus 15% employee discount
John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333, Hitachi VCR Model 2221 for $668
John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333, Sony VCR Model 321 for $999 on sale for $888.

Now put everything into a single file:

```
John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333,
Jane Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333
Kimo Smith, 23 Aloha Lane, Honolulu, HI 96822, 808 947-2222
John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333, Samsung VCR Model 555
for $299.95 minus 15% employee discount
John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333, Hitachi VCR Model
2221 for $668
John Doe, 46 Main St, Honolulu, HI 96822, 808 956-3333, Sony VCR Model 321
for $999 on sale for $888.
```

As you can see, we have all of the data, but none of the structure. What you can also hopefully see is that if you just start with data like this, it is not obvious what the table, field, and pointer structure should or could be; in fact there are many different structures that could be created around this data, which vary with respect to storage, performance, extendability, etc. For a complicated database, it is unlikely that two database designers would come up with exactly the same database structure given just a specification of the data to organize. It's similar to the fact that two programmers, given the specification of a program to write, would be unlikely to write the exact same code to solve the problem.

To summarize:

- Database structure is copyrightable expression, while the underlying data is not.
- It is possible to extract data from a database without retaining the copyrightable structure if you follow the approach I described above. If you do not, then you have extracted copyrightable structure along with the data.
- For a complex information system, there are many possible database structures that can be used to organize data.

It is my opinion that Fleming extracted both structure as well as the data from the Berry database, and thus are using a derived work. My opinion is based upon the following:

- Exhibits 7 through 18 show the use of "ID" fields present in the Berry database.

3. Exhibits used in forming this opinion:

- Exhibits 7 through 18

4. Qualifications.

Please see attached curriculum vitae.

5. Compensation:

Hourly rate: $200. Amount of retainer: $1000. Amount paid to date: $600 (3.0 hours)

6. Listing of cases in which the witness has testified

Berry v. Fleming Companies, Inc., Civ. No. 01-0446 SPK-LEK (D. Hawaii)

Signed: Philip Johnson

Date: 2/25/05